END

FILMED

Using the proposed heuristics and decomposing the system structure into
suitably sized modules allows simple and intuitive test procedures to be
developed with a minimum amount of computational effort.

References

Barlow, R. E. and F. Proschan (1975), "Importance of System Components and
Fault Tree Events", Stochastic Processes and Their Applications, Vol. 3,
pp.153-172.

Birnbaum, Z. W. (1969), "On the Importance of Different Components in a
Multicomponent System", Multivariate Analysis -- II, Ed: P. R. Krishnaiah,

INSPECTION POLICIES FOR FAULT LOCATION

by

DAVID A. BUTLER AND GERALD J. LIEBERMAN

TECHNICAL REPORT NO. 206

August 1982   TR-81

DEPARTMENT OF OPERATIONS RESEARCH
AND
DEPARTMENT OF STATISTICS
STANFORD UNIVERSITY
STANFORD, CALIFORNIA

INSPECTION POLICIES FOR FAULT LOCATION[1/]

by

David A. Butler
Oregon State University
and
Gerald J. Lieberman
Stanford University

0. Abstract

When a system fails it may not be obvious which components are at fault.
Locating faulty components to be replaced may require a series of inspections
each of which reveals the state (functioning/failed) of one of the components.
The order in which components are inspected and replaced can greatly affect the
cost to restore the system to an operating condition. This paper investigates
inspection sequences for complex coherent systems.

1. Introduction

When a system fails it is seldom obvious which components are at fault.
Locating the faulty parts may require a sequence of tests in each of which the
state (functioning/failed) of one of the components is identified. If the
state of every component must be determined, then the order in which components
are tested may not matter much. But often testing stops as soon as the first
failed component is found. Or testing may continue until a failed component is
located which, when replaced, fixes the system. (Such a component failure will

---

be called a _critical_ failure.) In these situations the average number of tests carried out depends upon the order in which components are tested.

Butterworth [1972] has developed two fault-testing models for k-out-of-n systems. In both models components fail independently of one another, each test costs a given amount, and test procedures are compared by their respective total expected costs. In the first model, the system is failed and test procedures must determine the state of each component. In the second model test procedures must determine the system state. Following Butterworth's nomenclature, a _sequential_ test procedure prespecifies the sequence in which components are tested; in a _nonsequential_ procedure, the outcomes of early tests may dictate the order of later tests. Sequential test procedures have an important advantage over nonsequential procedures in that they are much easier to specify and to implement. For determining the system state, Butterworth identifies a sequential procedure that is optimal among all procedures, sequential or not. For determining the component states, the optimal test procedure is sequential only for some special cases, among them parallel and series systems. For other k-out-of-n systems the optimal procedure seems to be neither easily identifiable nor easily implementable. For more complex systems the prospects are even worse. Our emphasis is, therefore, upon identifying good, if not optimal, test procedures for general coherent systems.

2. Sequential Test Procedures

The information required to specify a sequential test procedure is minimal -- just an ordered list of the system's components. In contrast, the amount of information necessary to specify a nonsequential policy may be quite large, because the choice of the nth component to be tested may depend arbitrarily

2

upon the 2^{n-1} possible results of the preceding $n-1$ tests. Unless there is a simple rule relating this choice to the previous test results, such a procedure might require an inordinate amount of information to be specified and, thus, be practically impossible to implement. For this reason we will focus our attention upon sequential test procedures.

Consider first the situation where testing stops as soon as a failed component is identified. In this case the optimal policy is sequential, because one knows in advance that all components tested prior to the last one will be working, and thus, the information known at any stage of testing can be prespecified. Let $\underline{\pi} = (\pi_1, \ldots, \pi_n)$ be a permutation of $1, \ldots, n$. The expected cost of the sequential policy determined by $\underline{\pi}$ is given by

$$F(\underline{\pi}) = c\left[1 + \sum_{k=1}^{n-1} \Pr\{\text{components } \pi_1, \ldots, \pi_k \text{ are functional at time of system failure}\}\right] \ .$$

Thus, in principle, if one can compute the above probabilities, one can determine the optimal test policy. In practice, however, these joint conditional failure probabilities are very hard to determine, and the number of permutations to be evaluated is very large.

An alternative is to choose a test policy heuristically. One way to do so is to compute the conditional failure probability of each component given system failure, and then to test the component having the highest conditional failure probability first, the one with the second highest conditional failure probability second, and so on. Although this procedure is not guaranteed to produce an optimal test policy, it is a reasonable heuristic.

3

Now consider the case where testing continues until a critically failed component is identified. Here testing often continues beyond the first detected failure, so the information known at any stage of testing cannot be specified in advance. Thus, the optimal policy is generally not sequential, and consequently, is that much harder to determine. The heuristic procedure suggested above may also be applied here, substituting critical failure probabilities in the selection criterion. As before, the policy it produces will not necessarily be optimal.

We now turn to the question of how to effectively compute the probabilities on which these heuristics are based.

3. Calculation of Conditional Failure Probabilities for Heuristic Procedures

Consider a coherent system (C, ϕ) of n components with minimal path sets $P_1, \ldots, P_r$. Assume that components fail independently of one another, and let $F_i(\cdot)$ be the time-to-failure distribution of component i. We assume a proportional hazards model for component failures, taking

$$\bar{F}_i(t) = e^{-\lambda_i R(t)} ,$$

where λ_i is a proportionality constant. A practical consequence of this assumption is that instead of specifying n different distribution functions, one need only specify a single hazard function together with $n - 1$ proportionality constants. In fact, the calculations in this paper will not even depend upon the hazard function but only the proportional hazards.

Let $\underline{p} = \bar{\underline{F}}(t) = (1 - F_1(t), \ldots, 1 - F_n(t))$ denote the vector of component reliabilities at a given time t. Let $h(\underline{p})$ be the system reliability

4

and let $I_h(i; \underline{p}) = \partial h(\underline{p})/\partial p_i$ be the Birnbaum reliability importance of component i. (Birnbaum [1969]).

Finally, let

$$A_h(i) = \text{Pr\{component } i \text{ is failed at time of system failure\}}$$

and

$$C_h(i) = \text{Pr\{component } i \text{ is critically failed at time of system failure\}} .$$

A quantity related to $A_h(i)$ and $C_h(i)$ is the Barlow-Proschan importance (Barlow and Proschan [1975]),

$$P_h(i) = \text{Pr\{component } i \text{ causes system failure\}} .$$

(A component is said to cause system failure if the failure of the system and the component coincide.) Identifying which component caused system failure cannot be determined by tests that only reveal whether or not a component is failed and nothing about how recently any failures may have occurred. Nonetheless, the causal probabilities can be computed. Barlow and Proschan [1975] give the following formulas for $P_h(i)$:

$$P_h(i) = \int_0^\infty I_h(i; \underline{\bar{F}}(t)) dF_i(t) .$$

In the case of proportional hazards, a simple change of variable reduces the formula to

5

$$P_h(1) = \int_0^1 I_h(1;\ \underline{p}^\lambda)\ \lambda_1 p^{\lambda_1 - 1}\, dp\ ,\qquad\qquad (1)$$

where

$$\underline{p}^\lambda = (p^{\lambda_1},\ p^{\lambda_2},\ \ldots,\ p^{\lambda_n})\ .$$

A similar formula can be developed for $A_h(1)$. Noting that $A_h(1)$ is the probability that the system functions at least as long as component 1 does,

$$A_h(1) = \int_0^\infty h(1_i,\ \bar{\underline{F}}(t))dF_1(t)$$

$$= \int_0^\infty h(1_i,\ \underline{p}^\lambda)\lambda_1 p^{\lambda_1 - 1}\, dp$$

$$= \int_0^\infty [h(\underline{p}^\lambda) + p_i I_h(1;\ \underline{p}^\lambda)]\lambda_1 p^{\lambda_1 - 1}\, dp\ .\qquad (2)$$

A comparable formula for $C_h(1)$ is more complicated. Suppose path sets $P_1,\ \ldots,\ P_k$ contains component 1 and $P_{k+1},\ \ldots,\ P_r$ do not. Let τ denote the coherent system with path sets $P_1 - \{1\},\ P_2 - \{1\},\ \ldots,\ P_k - \{1\}$ and let η denote the coherent system with path sets $P_{k+1},\ \ldots,\ P_r$. Component 1 is critically failed if 1) it causes system failure, or 2) when the system ulti- mately fails, there is a minimum path containing component 1 having the pro- perty that all other components are functioning. Thus,

$$C_h(1) = P_h(1) + \int_0^\infty \Pr\{X_1 < t\} \cdot \Pr\{\eta(\underline{X}) = t \text{ and } \tau(\underline{X}) > t\}dt\ .$$

Let $Z_i(s, t) = \Pr\{\eta(\underline{X}) < s \text{ and } \tau(\underline{X}) > t\}$ and let $z_i(t) = \dfrac{\partial}{\partial s} Z_i(s, t)\big|_{s=t}$. In the case of proportional hazards, $z_i(t)$ can be expressed as a polynomial in p, $g_i(p)$, and $C_h(i)$ reduced to

$$C_h(i) = P_h(i) + \int_0^\infty F_i(t) \cdot z_i(t)dt$$

$$= P_h(i) + \int_0^1 (1 - p^{\lambda_i}) \cdot g_i(p)dp \quad . \tag{3}$$

Equations (1), (2) and (3), in principle, allow the three failure-related pro-babilities $P_h(i)$, $A_h(i)$ and $C_h(i)$ to be calculated. However, these formulas may not be easy to evaluate directly. The integrands are polynomials in p, but determining the coefficients of these polynomials can be computa-tionally complex and very time consuming if the system is very large. Barlow and Proschan [1975] have suggested a Monte Carlo approach to approximate the integral in (1). A similar procedure could be used to evaluate (2) and (3). As an alternative, we will develop an analytic methodology which exploits the existance of modules within the coherent system to facilitate the computations.

A module of a coherent system $(C, \emptyset)$ is a subset A of components organized into some coherent substructure χ such that the system performance depends only upon the components in A through the performance of χ, i.e.,

$$\emptyset(\underline{x}) = \psi(\chi(\underline{x}^A), \underline{x}^{A^c}) \quad ,$$

7

where $\underline{x}^A$ denotes a vector with components x_i, $i \varepsilon A$, and $A^C = C - A$. The structure function ψ is called the organizing structure. Birnbaum [1969] shows that the overall importance of a component within a module is the product of its importance within the module and the importance of the module within the organizing structure, i.e.,

$$I_h^\phi(i) = I_h^\chi(i) \cdot I_h^\psi(M) \quad . \tag{4}$$

This formula allows the computation of the integrand in (2) to proceed on a module-by-module basis instead of having to consider the entire system at once. This is a crucial advantage, since the computations involved in determining $h(\underline{p})$, $I_h(i; \underline{p})$, or $g_i(p)$ directly grow exponentially with the number of min paths.

In order to calculate $C_h(i)$ efficiently, a modular decomposition formula for $g_i(p)$ is needed.

Proposition:

$$g_i^\phi(p) = g_i^\chi(p) \cdot I_h^\phi(M; p) + I_h^\chi(i; p) \cdot g_M^\phi(p) \tag{5}$$

Proof: Suppose ψ is defined by min paths $P_1, \ldots, P_s$. Suppose $P_1, \ldots, P_\ell$ contain module M and $P_{\ell+1}, \ldots, P_s$ do not. Let τ_M^ϕ denote the structure function defined by min paths $P_1 - \{M\}, \ldots, P_\ell - \{M\}$, and let η_M^ϕ be the

8

structure function defined by $P_{\ell+1}, \ldots, P_s$. Define τ_i^{ϕ}, η_i^{ϕ}, τ_i^{χ}, and η_i^{χ} similarly. These quantities are interrelated as follows:

$$\tau_i^{\phi} = \tau_M^{\chi} \cdot \tau_i^{\chi}$$

$$\eta_i^{\phi} = \eta_M^{\Psi} + \tau_M^{\Psi} \cdot \eta_i^{\chi} - \eta_M^{\Psi} \cdot \tau_M^{\Psi} \cdot \eta_i^{\chi} \; .$$

Let T_i denote the lifetime of component i, and let

$$X_i(t) = \begin{cases} 1 & \text{if } T_i \geq t \\ 0 & \text{otherwise} \; . \end{cases}$$

$$Z_i^{\phi}(s, t) = \Pr\{\eta^{\phi}(\underline{X}(s)) = 0 \quad \text{and} \quad t_i^{\phi}(X(t)) = 1\}$$

$$= \Pr[\{\eta_M^{\Psi}(\underline{X}(s)) = 0\} \quad \text{and} \quad \{t_M^{\Psi}(\underline{X}(s)) = 0 \quad \text{or} \quad \eta_i^{\chi}(\underline{X}(s)) = 0\}$$

$$\text{and} \quad \{t_M^{\Psi}(\underline{X}(t)) = 1\} \quad \text{and} \quad \{\tau_i^{\chi}(\underline{X}(t)) = 1\}] \; .$$

For $s \leq t$,

$$Z_i^{\phi}(s,t) = \Pr\{\eta_M^{\Psi}(\underline{X}(s)) = 0 \quad \text{and} \quad t_M^{\Psi}(\underline{X}(t)) = 1\}$$

$$\cdot \Pr\{\eta_i^{\chi}(\underline{X}(s)) = 0 \quad \text{and} \quad t_i^{\chi}(\underline{X}(t)) = 1\}$$

$$= Z_M^{\Psi}(s, t) \cdot Z_i^{\chi}(s, t) \; .$$

$$z_i^{\phi}(t) = z_M^{\Psi}(t) \cdot Z_i^{\chi}(t, t) + Z_M^{\Psi}(t, t) \cdot z_i^{\chi}(t) \; .$$

Since for any coherent structure,

$$Z_i(t, t) = \Pr\{\eta(\underline{X}(t)) = 0 \quad \text{and} \quad \tau(\underline{X}(t)) = 1\}$$

$$= h(1_i, p^{\underline{\lambda}}) - h(0_i, p^{\underline{\lambda}})$$

$$= I_h(i, p^{\underline{\lambda}}) \quad ,$$

the result is established.

$\square$

4. Computational Notes and an Example

A computer program to determine conditional-failure and critical-failure probabilities has been developed by and is available from the authors. This program is written in ANSI-Standard FORTRAN and exploits modules to efficiently compute $A_h(i)$ and $C_h(i)$. The program only allows integer values for the λ_i. It is easy to show that if one multiplies each λ_i by a common positive constant no change to $A_h(i)$ or $C_h(i)$ results. Thus, requiring integer values for the λ_i is not a very severe restriction. The system structure is entered in a "bottom-up" fashion, with the most elementary modules input first. The inclusion-exclusion method is used to compute the reliability $h(\underline{p})$ of each module immediately after it is input. As each $h(\underline{p})$ is computed, $I_i(\underline{p})$ and $g_i(p)$ are determined also. As modules are input, equations (4) and (5) are used to determine the overall importance and critical polynomials. Once all modules have been input, equations (2) and (3) are used to compute $A_h(i)$ and $C_h(i)$. Thus, the heuristic sequential inspection policies for the two optimization criteria may be obtained. Special methods of encoding polynomials and of sequencing through path subsets are employed to improve speed.

10

To illustrate the efficiency of exploiting modules in determining these failure probabilities, consider the following example system.
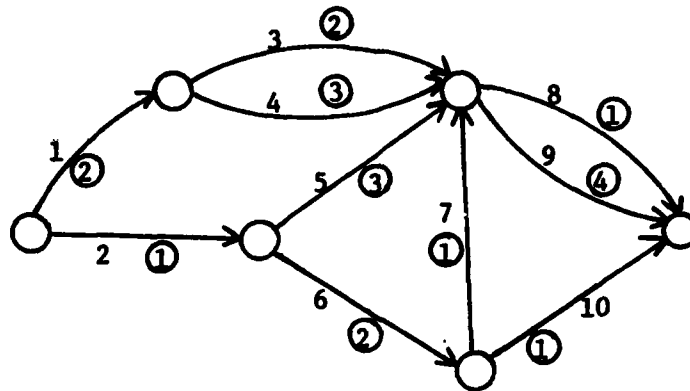
The circled numbers are the failure rates. The table below gives the conditional-failure and critical-failure probabilities.

i	$A_h(i)$	$C_h(i)$
1	0.605	0.350
2	0.385	0.292
3	0.568	0.213
4	0.690	0.213
5	0.729	0.380
6	0.635	0.439
7	0.336	0.046
8	0.368	0.238
9	0.777	0.238
10	0.350	0.122

Optimization Criterion	Heuristic Inspection Sequence
first failure	9, 5, 4, 6, 1, 3, 2, 8, 7, 10
first critical failure	6, 5, 1, 2, 8, 9, 3, 4, 10, 7

11

To compute the above directly (without breaking down the system into modules) required 10.7 CPU-seconds on a CYBER 73. The most extensive modular decomposition of the system would be to consider components 3 and 4 as one module, component 1 and the first module as a second module, components 8 and 9 as a third module, and components 2, 5, 6, 7, 10 and modules 2 and 3 as a fourth module. Utilizing this modular decomposition reduced the computation time by 96% — to 0.4 CPU-seconds.

Although this computer program was developed on a large computer, it should be easily adaptable to microcomputers. Almost all of the calculations involve integer arithmetic (assuming integer values for the λ_i). Only the final polynomial integrations require floating point division. Overall memory requirements can be large, because each component has associated with it two polynomials, $I_h(i; p^{\lambda})$ and $g_i(p)$, each of which may have hundreds of terms. In addition, each module has associated with it a reliability polynomial, $h(p^{\lambda})$. However, only small portions of this information are needed at any given time, and so it can be stored on relatively slow-speed media, such as floppy disks.

5. Conclusions

Locating faulty components in a failed system may require costly inspections. The total expected cost to find a failed or critically failed component can be significantly influenced by the order in which components are tested. Determining the optimal inspection sequence is impractical for all but the simplest system structures, so heuristic procedures must be used.

Using the proposed heuristics and decomposing the system structure into suitably sized modules allows simple and intuitive test procedures to be developed with a minimum amount of computational effort.

References

Barlow, R. E. and F. Proschan (1975), "Importance of System Components and Fault Tree Events", Stochastic Processes and Their Applications, Vol. 3, pp.153-172.

Birnbaum, Z. W. (1969), "On the Importance of Different Components in a Multicomponent System", Multivariate Analysis -- II, Ed: P. R. Krishnaiah, Academic Press, New York.

Butterworth, R. (1972), "Some Reliability Fault-Testing Models", Operations Research, Vol. 20, No. 2, pp.335-343.
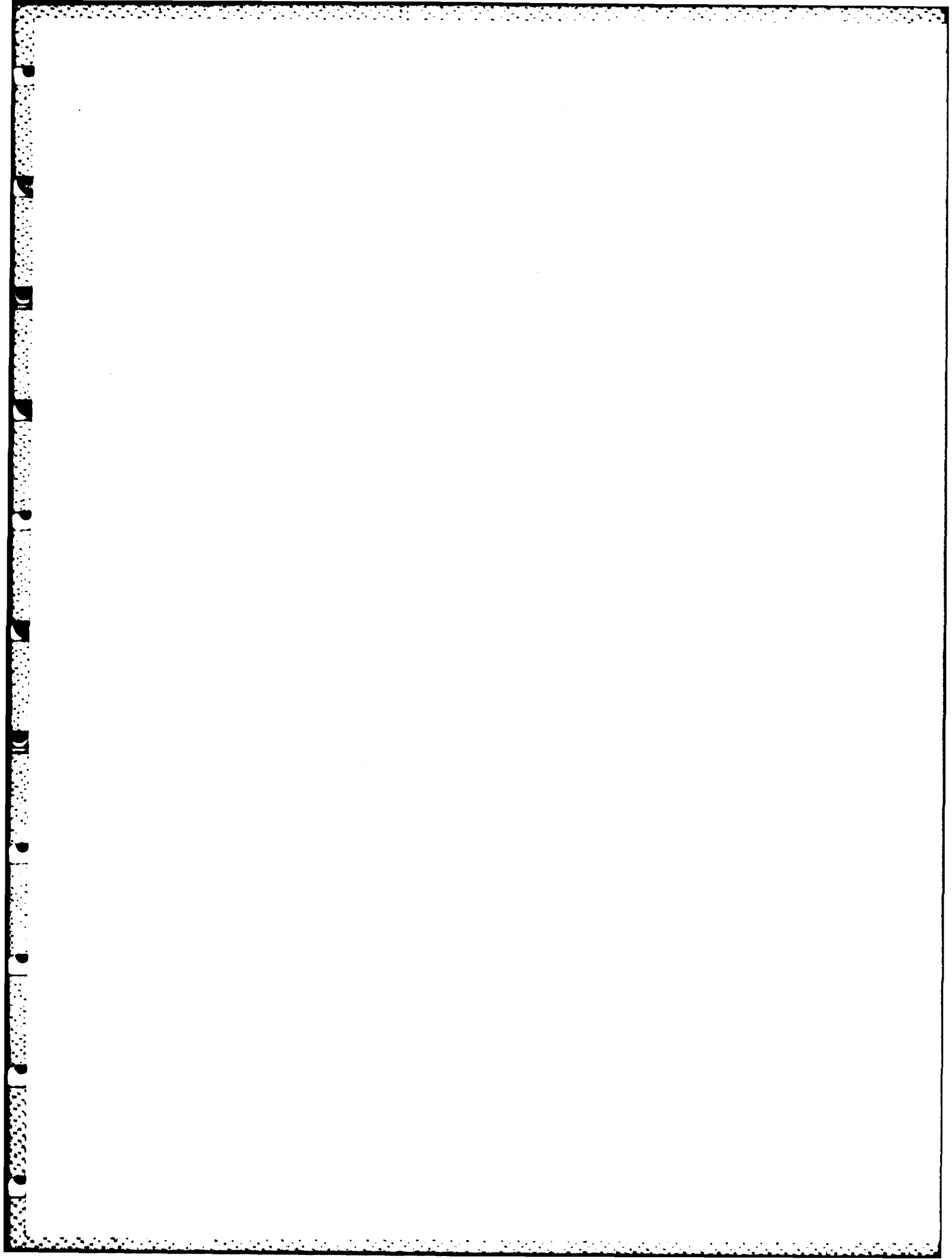
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR — 206	2. GOVT ACCESSION NO. AD-A120661	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) INSPECTION POLICIES FOR FAULT LOCATION		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) David A. Butler and Gerald J. Lieberman		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0561
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Operations Research and Department of Statistics - Stanford University, Stanford, California 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (NR-047-200)
11. CONTROLLING OFFICE NAME AND ADDRESS Operations Research, Code 434 Office of Naval Research Arlington, Virginia 22217		12. REPORT DATE August 1982
		13. NUMBER OF PAGES 13
14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

FAULT LOCATION

INSPECTION POLICIES

HEURISTIC PROCEDURES

FAILED COMPONENT

CRITICALLY FAILED COMPONENT

SYSTEM DECOMPOSITION

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(see next page)

DD FORM 1473 JAN 73   EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Abstract: INSPECTION POLICIES FOR FAULT LOCATION   Report #206

When a system fails it may not be obvious which components are at fault. Locating faulty components to be replaced may require a series of inspections each of which reveals the state (functioning/failed) of one of the components. The order in which components are inspected and replaced can greatly affect the cost to restore the system to an operating condition. This paper investigates inspection sequences for complex coherent systems.

END

FILMED